

Towards edge-cloud computing

Tianfield, Huaglory

Published in:
2018 IEEE International Conference on Big Data

DOI:
[10.1109/BigData.2018.8622052](https://doi.org/10.1109/BigData.2018.8622052)

Publication date:
2019

Document Version
Author accepted manuscript

[Link to publication in ResearchOnline](#)

Citation for published version (Harvard):
Tianfield, H 2019, Towards edge-cloud computing. in *2018 IEEE International Conference on Big Data*. IEEE, pp. 4883-4885, 2018 IEEE International Conference on Big Data , Seattle, United States, 10/12/18.
<https://doi.org/10.1109/BigData.2018.8622052>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please view our takedown policy at <https://edshare.gcu.ac.uk/id/eprint/5179> for details of how to contact us.

Towards Edge-Cloud Computing

Huaglory Tianfield

Department of Computing, Glasgow Caledonian University

Cowcaddens Road, Glasgow G4 0BA, United Kingdom

e-mail: h.tianfield@gcu.ac.uk

Abstract—In this paper, we propose edge-cloud computing as a new framework for organizing the data pipeline functions, i.e., acquisition, processing and analytics, in the Internet of Things systems. Edge-cloud computing consists of three layers, namely smart edge devices, fog computing layer and cloud computing layer. Being an intrinsically hierarchical framework, edge-cloud computing is able to tackle the complexities of Internet of Things systems more fundamentally. In particular, we discuss the main roles of orchestration and intermediary that fog computing undertakes for microservices and cloud services, and the fundamental underpinning function – automated service provisioning in fog computing.

Keywords—cyber-physical cloud computing, fog computing, Internet of Things, service provisioning, service orchestration.

I. INTRODUCTION

The Internet of Things (IoT) concept refers to the devices that are connected to each other. IoT permeates everywhere and has widespread applications. The devices connected can include anything, whether in industrial environments or in human activity environments, e.g., sensors, actuators, wearable devices, connected vehicles, augmented reality kits, game consoles, robots, drones, Autonomous Underwater Vehicles (AUV), Unmanned Aerial Vehicles (UAV), and so forth.

IoT has three prominent characteristics, that is,

- a) IoT is a mission-oriented real-time system;
- b) IoT system is data intensive, or even data explosive;
- c) IoT system has very constrained on-board/on-site processing capability.

Traditional architectures for IoT systems are mostly centralized where the data pipeline functions, i.e., data acquisition, data processing and data analytics, are coordinated in a fundamentally centralized fashion. Centralized architectures of IoT systems inevitably suffer from data latency and system scalability issues.

To address the data intensiveness and on-site resource constraint of IoT systems, very readily it turns to cloud computing. This leads to the concept of cyber-physical cloud computing [1]. However, introducing cloud computing into IoT systems will not automatically resolve the data latency and system scalability issues. Just the opposite, cloud computing can actually deteriorate the issues, especially data latency. This is then where fog computing is much expected

of [2] [3]. The purpose for introducing fog computing into IoT systems seems to have been highlighted on moving the cloud computing near where the devices are located and where the data are generated, eliminating the data latency caused by having to transmit a large amount of data for processing in a remote cloud datacenter.

Although localizing cloud computing in the proximity of the edge devices and the data sources seems to make sense, it is quite confusing conceptually in that the fog computing is defined as another type of cloud computing, e.g., a lightweight cloud computing. One of the problems is that what fog computing is supposed to perform is actually not something of a cloud computing, and none of the models and enabling technologies of cloud computing can be meaningfully moved to the fog computing. For example, the very basic concept of cloud computing is that everything is delivered as a service, but with the fog computing, what is delivered as a service and what are the enabling technologies underpinning the fog computing? Essentially, cloud computing is about computing, whereas IoT is a mission-oriented real-time system rather than computing.

In this paper, we take a fundamental view on the complexities of IoT systems. It is widely observed that hierarchy is the most powerful approach to tackling system complexities. We will put forward edge-cloud computing as an intrinsically hierarchical framework for organizing the data pipeline functions of IoT systems. Edge-cloud computing is not any other type of cloud computing. Rather, edge-cloud computing is a unified framework for IoT systems in which smart edge devices, fog computing and cloud computing coexist.

II. HIERARCHICAL FRAMEWORK OF EDGE-CLOUD COMPUTING

Basically, our proposed hierarchical framework for IoT systems, called edge-cloud computing, consists of three layers, namely smart edge devices, fog computing layer, and cloud computing layer, as illustrated in Fig. 1.

Furthermore, within each layer of edge-cloud computing, it is possible to employ a local mesh structure. For example, the connections among the smart edge devices may employ wireless mesh networks. Smart edge devices are grouped into clusters, each group has a cluster head, which is a mesh router, and cluster head will link via a smaller number of gateways to fog computing.

Similarly, fog computing may employ an internal mesh structure. Fog nodes are grouped and each group has a fog head. Fog nodes mainly deal with the cluster heads of the mesh networks of the smart edge devices, whereas fog heads

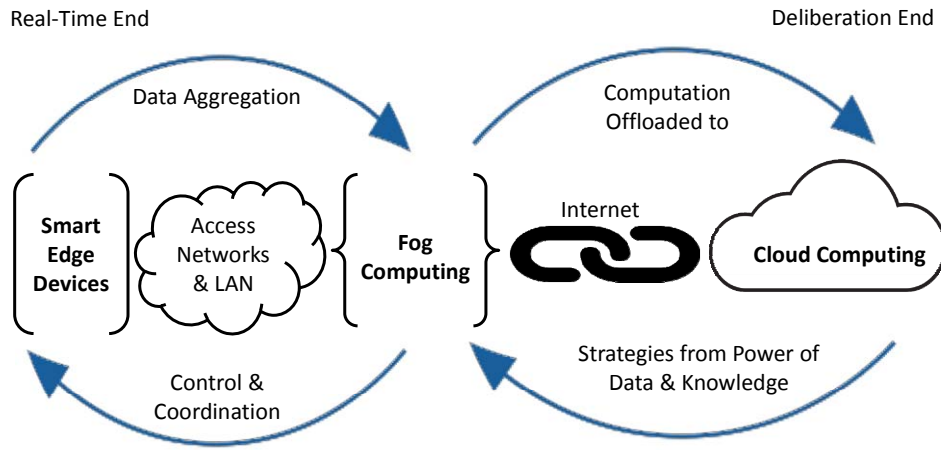


Fig. 1. Hierarchical framework of edge-cloud computing

coordinate fog nodes and liaise with the cloud computing layer.

In the hierarchical framework of edge-cloud computing, fog computing lies at the pivotal middle layer between smart edge devices and the cloud computing layer, and assumes the roles of orchestration and intermediary for microservices, from the smart edge devices, and cloud services, from the cloud computing layer, thus conciliating and synchronizing the far differing dynamisms at real-time scale and deliberation scale, as illustrated in Fig. 2.

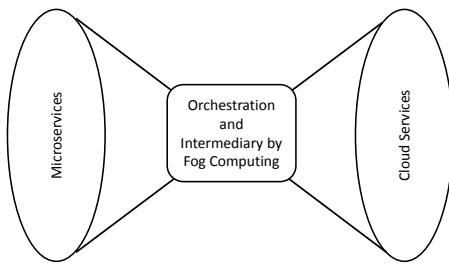


Fig. 2. Fog computing assumes orchestration and intermediary for microservices and cloud services

III. AUTOMATED SERVICE PROVISIONING

As stated above, the main roles of fog computing have to undertake are orchestration and intermediary for microservices and cloud services. Automated service provisioning is the fundamental function to realize such roles.

Service provisioning lies at the middle of the service lifecycle, as illustrated in Fig. 3.

The upper and lower sections of the curved block arrow represent the upstream and the downstream of a service lifecycle, respectively. Lifecycle is the growth stages that the same service undergoes. However, there is a subtle difference between upstream and downstream, that is, in the upstream, the service is in its generic form, whereas in the downstream, as a result of the service provisioning process, the service becomes a specific instance of the service, which has been customized to meet the service request specifications.

According to the technical group that created the Services Provisioning Markup Language (SPML), provisioning is the

automation of all the steps required to manage (setup, amend, and revoke) user or system access entitlements or data relative to electronically published services [4].

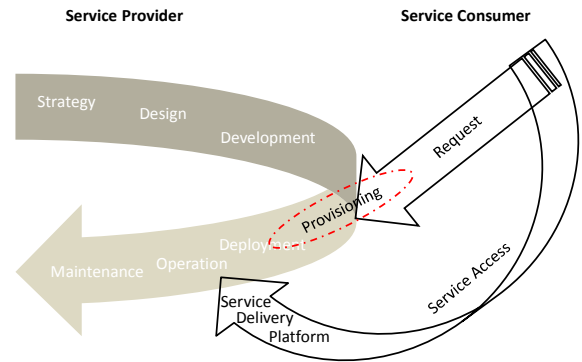


Fig. 3. Service lifecycle

What the service provisioning performs is to leverage the provider's supply and the consumer's demand and then work out how the service should be provisioned in terms of some criteria, e.g., minimum costs, optimal resource utilization, lowest downtime, minimum energy consumption, etc.

In particular, service provisioning performs 4 main functions, i.e.,

- 1) Leveraging demand and supply
 - Demand: service request from user/consumer
 - Supply: service description from provider
- 2) Formation of service-level agreement (SLA)
- 3) Arrangement for service deployment
 - Systems configurations
 - Service execution schedule
- 4) Arrangement for service delivery
 - Identity provisioning
 - Identity and access management policy

Service provisioning process takes as its inputs:

- (a) what services are available from the supply by the service provider, and

(b) what service is demanded by the service consumer; and then works out its outputs, i.e., how services should be provisioned.

At a very abstract level, main inputs and outputs of service provisioning process can be illustrated as in Fig. 4.

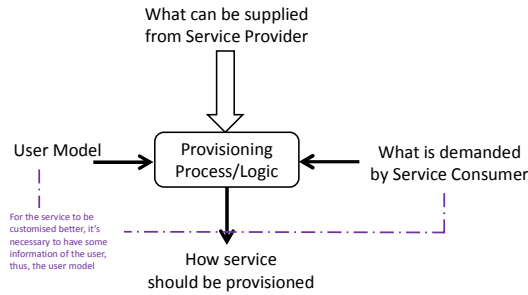


Fig. 4. Inputs and output of service provisioning

More specifically, main inputs of service provisioning process can be elaborated as follows.

- 1) What services can be supplied by the service provider. Once the provider has had a service ready, it will produce a complete specification document called *service description*, and then publish the service description in to an indexing or directory, say, the service catalog. So, put in simple words, the first input is the service catalog about all the services that are currently available to supply;
- 2) What is demanded by the service consumer. The detail of what the consumer wants is represented as a specification document called *service request*. So, the second input is the service request from the consumer;
- 3) Service provisioning will involve customizing a generic service into a specific instance of the service that suits the service request. For the service to be customized better, it is necessary to have some information of the user, i.e.,

the user model. Therefore, the third input is the user model. With a user model (including user identity) established, it will also help the provider to authenticate the legitimate user and authorize the user to the right access to the service.

Main outputs of service provisioning process are as follows.

- 1) Identity and Access Management (IAM) policies,
- 2) Service-Level Agreement (SLA),
- 3) The schedule for access to the services/resources, as calculated by the resource scheduling and allocation algorithms.

IV. SUMMARY

We have taken a very fundamental view that fog computing is not another type of cloud computing. Rather, fog computing lies at the pivotal middle layer between smart edge devices and the cloud computing layer, and undertakes the roles of orchestration and intermediary for the microservices and the cloud services. The new formwork as proposed, i.e., edge-cloud computing is holistic for organizing the data pipeline functions, i.e., data acquisition, data processing and data analytics in the whole IoT system and is an intrinsically hierarchical framework, which would be able to tackle the complexities of IoT systems more fundamentally.

REFERENCES

- [1] E. Simmon, et al., A vision of cyber-physical cloud computing for smart networked systems, NISTIR 7951, August 2013, 61 pages.
- [2] M. Iorga, et al., Fog computing conceptual model, NIST Special Publication 500-325, March 2018, 14 pages.
- [3] Fog Computing https://en.wikipedia.org/wiki/Fog_computing
- [4] G. Cole, et al., OASIS Service Provisioning Markup Language (SPML), Version 2, OASIS Standard, April 2006.